TRANSMISIA SERIALA A DATELOR. TRANSMISIA SINCRONA. STANDARDUL ETHERNET. SISTEME INCORPORATE CONECTATE LA INTERNET

1. Transmisia sincrona a datelor.

1.1 Transferul datelor prin mesaje.

Utilizarea mesajelor este preferata in cazul in care se urmareste transmiterea unei cantitati mai mari de date cu un protocol de comunicatie mai sofisticat (de ex. controlul erorilor). Transferul de mesaje este modul de comunicare a informatiei cel mai intalnit in cazul transmisiilor seriale intr-o retea de date (dispozitive).

Structura mesajului

In ultima instanta mesajul reprezinta o multime de biti care trebuie transferati de la un punct la altul (fig. 1).

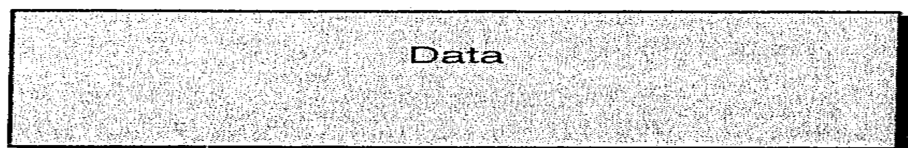


Fig. 1 – Mesajul util

Pentru a putea controla aceasta transmisie trebuie aplicate reguli in cea ce priveste transmisia si interpretarea bitilor. Astfel structura mesajului poate fi detaliata ca in fig.2,

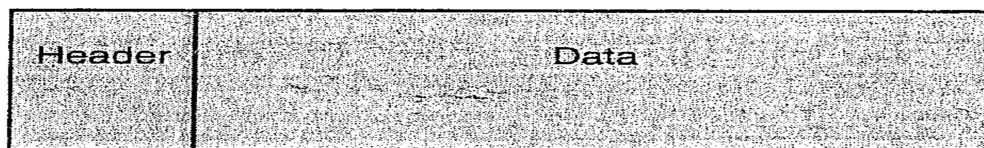


Fig. 2 – Adaugarea informatiei de control

in care bitii sunt grupati in doua componente: mesajul util (bitii de date, payload) si informatia de control (bitii de tip 'antet' (header)) care este adaugata (de driverul de comunicatie) pentru a facilita transmiterea mesajului util, informatie care contine in mod obisnuit informatie de adresa si informatie pentru controlul si sincronizarea transmisiei.

Nu toate mesajele au aceeasi dimensiune, aceasta putand varia de la cativa octeti pana la blocuri de date de lungime variabila. Transmisia mesajelor devine mai clara si mai robusta daca dimensiunea mesajului (blocului de date) transmis la un moment dat (care poate constitui o parte a mesajului initial) este aceeasi. Aceste blocuri de date sunt cunoscute de obicei sub denumirea de pachete sau datagrame. Aceasta poate presupune in

unele cazuri ‘umplerea’ blocului de date pana la dimensiunea prestabilita (corespunzatoare). Structura mesajului impartit in pachete este prezentata in fig. 3.

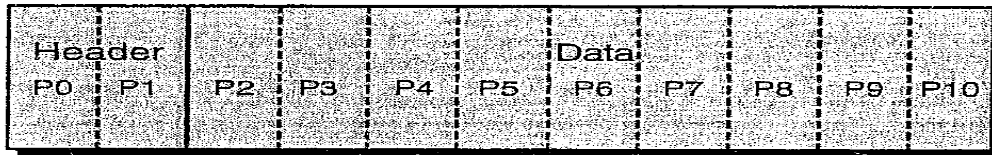


Fig. 3 – Impartirea mesajelor in pachete.

In final fiecare pachet primeste o informatie de adresa ca in fig. 4.

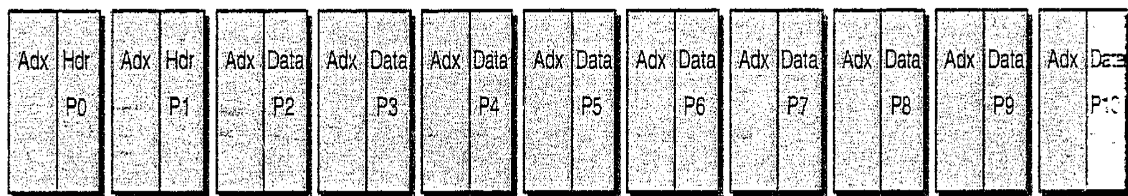


Fig. 4 – Pachetul care pleaca de la emitator.

Receptorul trebuie sa fie capabil sa accepte fluxul de informatie care este trimis spre el si sa detecteze si sa identifice inceputul si sfarsitul unui bit, inceputul si sfarsitul unui caracter precum si inceputul si sfarsitul unui bloc de date sau cadru de date (frame). Este ceea ce se numeste sincronizare la nivel de bit, de caracter sau de cadru de date.

Antetul.

Este adaugat de catre driverul de comunicatie pe baza cerintelor protocolului de comunicatie de la fiecare nivel (al arhitecturii stratificate). Antetul poate cuprinde:

- Adresa de destinatie (prezenta totdeauna).
- Adresa sursei.
- Indentificator pentru natura informatiei din mesaj:
  - Mesaj de comanda (control)
  - Mesaj de date, etc.
- Informatie privind calea pe care mesajul (pachetul) trebuie sa o urmeze pentru a ajunge la destinatie.
- Identificator pentru lungimea (dimensiunea) mesajului:
  - Identificatori de inceput si de sfarsit de mesaj
  - Identificator de inceput de mesaj impreuna cu identificator pentru lungimea mesajului (*length field*), etc.
- Informatie pentru verificarea corectitudinii transmisiei (*error management field*):
  - Detectie erori - cel mai simplu caz: bitul de paritate.
  - Detectie si corectie erori – campuri CRC (Cyclic Redundancy Check).

Controlul si sincronizarea sunt realizate in doua moduri: prin structura antetului mesajului si prin schema de transmisie (protocolul de comunicatie).

Schimbul de date se face prin intermediul unei interfete ‘inchise’ (proprietatea cuiva), unui standard modificat cum ar fi EIA 232 sau prin intermediul unei versiuni dedicate al unor standarde cum ar fi USB sau Firewire.

1.2 Transmisia sincrona.

In transmisia sincrona schimbul de date intre emitator si receptor este realizat in mod sincron sub controlul unui ceas in mod direct (semnal de ceas transmis in paralel cu datele) sau prin codificarea (includerea) semnalului de ceas in date.

In fig. 5 si 6 se prezinta cazul a doua protocoale/standarde de transmisie sincrona in care ceasul este trimis separat insotind semnalul de date.

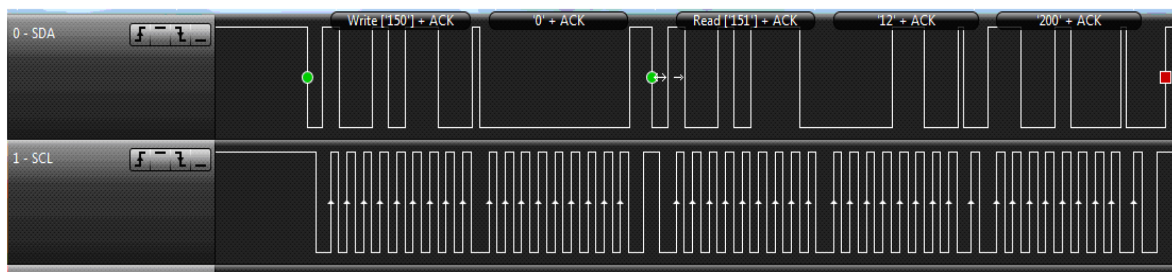


Fig. 5 – I2C.   SDA date; SCL ceas

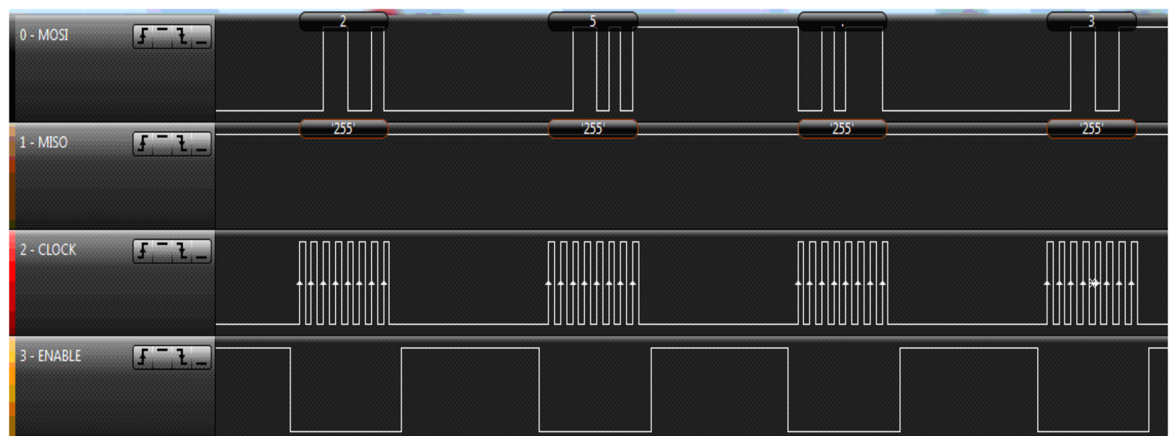


Fig. 6 – SPI,.   MOSI = Master Output, Save Input (date).

Si in cazul transmisiei sincrone trebuie realizata realizata sincronizarea pe bit, caracter si cadru de date. In general sincronizarea pe cadru este derivata din sincronizarea la nivel de bit si din cea la nivel de caracter.

Sincronizarea la nivel de bit

In cazul codificarii semnalului de ceas in date sincronizarea la nivel de bit se realizeaza in doi pasi:

1. Se codifica semnalul de ceas in date.
2. Se reconstruieste semnalul de ceas din date.

Pentru codificarea semnalului de ceas in date se folosesc mai multe metode dintre care amintim:

1. Codificare bipolara, ca in fig. 7. Semnalele logice (0, 1) sunt reprezentate prin polaritatea diferita a semnalelor. Fiecare celula de bit contine informatie de ceas reflectata de catre tranzitia din centrul fiecarei celule de bit. Se observa de asemenea ca semnalul (codificat) se intoarce la zero dupa fiecare bit codificat. Un astfel de algoritm se numeste algoritm de tip reintoarcere la zero (Return-to-Zero - RZ),

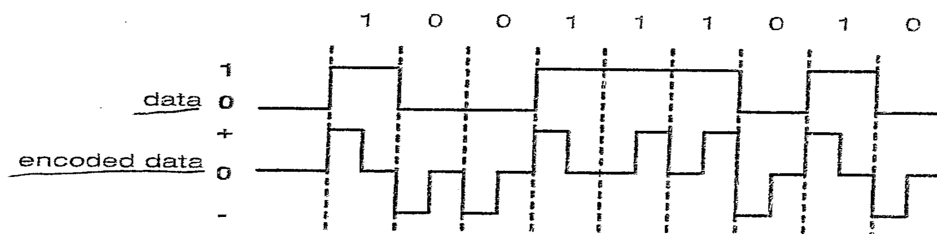


Fig. 7 – Codificare bipolara.

Se observa de asemenea ca metoda necesita 3 nivele diferite de semnal (+, 0 si -).

2. Codificare (de faza) Manchester (Ethernet), ca in fig. 8.

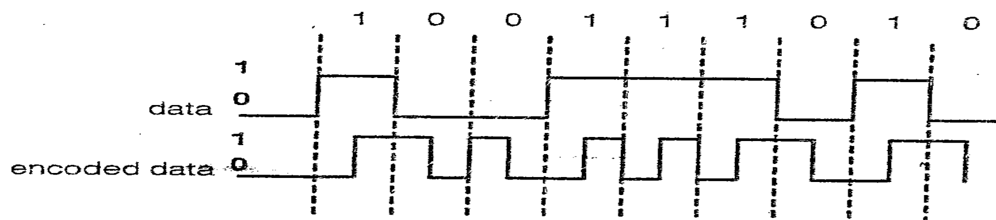


Fig. 8 – Codificare Manchester de faza.

In aceasta codificare fiecare bit de zero este codificat printr-o tranzitie de la 1 la 0, iar un bit de 1 este codificat ca o tranzitie de la 0 la 1 (IEEE 802.3 – Ethernet, IEEE 802.4 – Token Bus). Tranzitiile apar in centrul celulei de bit furnizand astfel informatii despre semnalul de ceas. Se observa de asemenea ca semnalul (codificat) nu se mai intoarce la nivelul 0 dupa codificarea fiecarui bit. Un astfel de algoritm se numeste algoritm de tip fara intoarcere la zero (Non-Return-to-Zero), sau NRZ.

Obs: Codificarea Manchester de faza poate fi vazuta ca un caz particular de modulatie BPSK.

3. Codificare Manchester diferentiala, (reteaua LON) ca in fig. 9
In aceasta codificare exista o tranzitie la inceputul fiecarui bit numai daca urmatorul bit de codificat este 0 logic.

Obs: Codificarea Manchester diferentiala poate fi vazuta ca un caz particular de modulatie PSK.

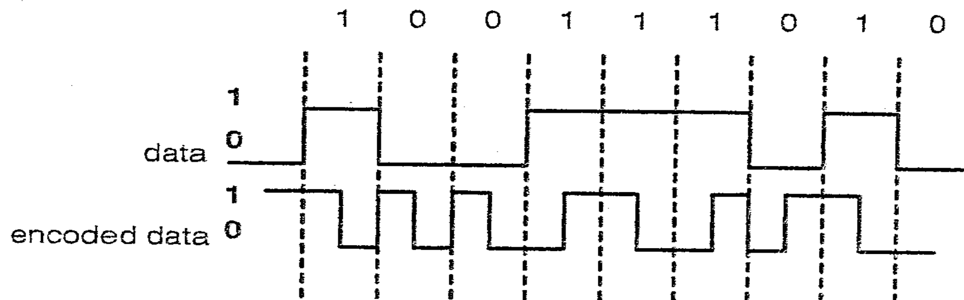


Fig. 9 – Codificare Manchester diferentiala.

4. Codificare NRZI.
In aceasta codificare exista o tranzitie numai daca se transmite un ‘0’ (sau un ‘1’. In cazul magistralei/protocolului USB este valabila prima varianta).

Codificarea este pusa in evidenta de fig. 10.

NRZI (1)

- Want to transmit :
  =>1 : without changing the level of the level of the signaling
  =>0 : flip the value of the differential pair
  example :

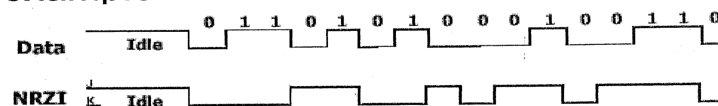


Fig. 10 - Codficare NRZI (USB).

Obs: Semnalul de ceas este mai dificil de recuperat din semnalele codificate NRZI fiind posibila pierderea de date in cazul unor secvente particulare de biti. Pentru a evita acest lucru se utilizeaza mai multe tehnici, una dintre cele mai cunoscute fiind cea denumita ‘umplerea cu/adaugarea de biti’ (bit stuffing) – magistrala USB. Astfel, dupa un sir de 6 biti de ‘1’ este obligatorie introducerea unui bit de ‘0’ (putand astfel genera viteze de transmitere variabile), ca in fig. 11.

NRZI (2)

- Problem :
  When we send "1" stream , the transmission line will stay static (no change period)
- Solution :
  "Bit stuffing" , performed before the NRZI
  example :
  data : 010**1111111**01
  send : 010**11111110**101

Fig 11 – Umplere cu biti (bit-stuffing - USB).

<u>Refacerea ceasului.</u>

Pentru a putea reface semnalul de ceas din datele receptionate fiecare transmisie incepe cu un <u>preambul</u> care include informatie necesara sincronizarii (<u>secventa de sincronizare</u>). Pentru refacerea ceasului si esantionarea corecta a informatiei din linia de comunicatie se pot utiliza la receptie circuite de tip PLL (phase-locked-loop) bazate pe un semnal de ceas foarte stabil. Codificarea datelor trebuie facuta astfel incat sa existe un numar suficient de tranzitii in semnalul de date Ceasul de esantionare este ajustat la fiecare tranzitie pentru a se asigura esantionarea la mijlocul fiecarui bit.. Schema bloc principiala a unui circuit PLL este prezentata in fig. 12.

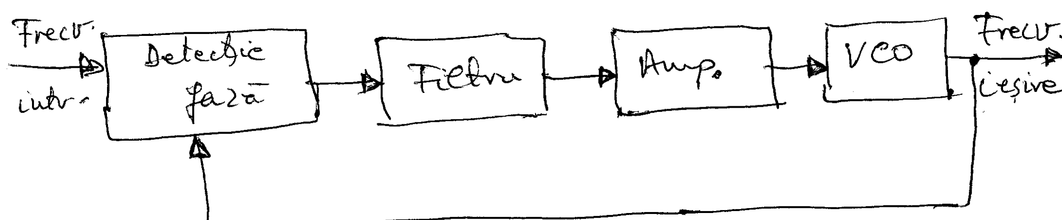


Fig. 12 – Circuit PLL.

Iesirea circuitului PLL este generata cu ajutorul unui oscilator controlat intensiune (Voltage-Controlled Oscillator, VCO). Iesirea VCO este intoarsa la intrarea circuitului intr-un bloc de detectie a fazei care compara caracteristicile semnalului de intrare si ale celui de iesire. Cand diferenta intre frecventa si faza celor doua semnale este zero se spune ca sistemul s-a 'blocat' (fixat, locked) pe frecventa de intrare.

Semnalul de iesire al PLL este folosit pentru esantionarea fiecarui bit din linia de transmisie. Semnalul de intrare este oricare dintre semnalele codificate sosit pe linia de transmisie, parte dintre ele fiind descrise anterior.

<u>Probleme care pot aparea la transmisia sincrona:</u>

1. Dispozitivele trebuie sa functioneze cu aceeasi viteza.

2. Viteze de propagare diferite a semnalelor de date si de ceas (incarcari diferite ale liniilor de date si de ceas).
3. Exista posibilitatea ca semnalele de ceas sa soseasca la destinatii diferite la momente de timp diferite ca in fig. 13.

Aceasta inseamna aparitia fenomenului de clock-skew (in raport cu datele) in cazul magistralelor lungi si a vitezelor mari ale semnalului de ceas.

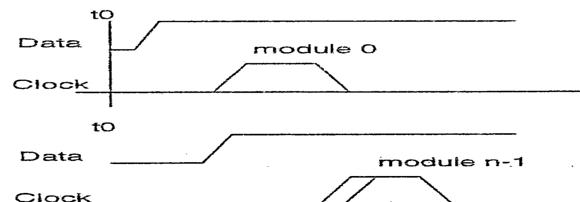


Fig 13 – Desincronizarea/distorsionarea ceasului (clock-skewing).

Avantaje ale transmisiei sincrone:
1. Mai usor de testat.
2. Protocolul este mai simplu decat la transmisia asincrona.
3. Mentinerea sincronizarii este mai usoara decat la transmisia asincrona.

2. Standardul Ethernet.

Pentru reteaua Ethernet se folosesc cabluri coaxiale si cabluri bifilare torsadate. Tipurile de cabluri bifilare utilizate sutn prezentate in fig. 14.

Name	Typical construction	Bandwidth	Applications	Notes
Level 1		0.4 MHz	Telephone and modem lines	Not described in EIA/TIA recommendations. Unsuitable for modern systems.[7]
Level 2		4 MHz	Older terminal systems, e.g. IBM 3270	Not described in EIA/TIA recommendations. Unsuitable for modern systems.[7]
Cat.3	UTP[8]	16 MHz[8]	10BASE-T and 100BASE-T4 Ethernet[8]	Described in EIA/TIA-568. Unsuitable for speeds above 16 Mbit/s. Now mainly for telephone cables[8]
Cat.4	UTP[8]	20 MHz[8]	16 Mbit/s[8] Token Ring	Not commonly used[8]
Cat.5	UTP[8]	100 MHz[8]	100BASE-TX & 1000BASE-T Ethernet[8]	Common in most current LANs[8]
Cat.5e	UTP[8]	100 MHz[8]	100BASE-TX & 1000BASE-T Ethernet[8]	Enhanced Cat5. Same construction as Cat5, but with better testing standards.
Cat.6	UTP[8]	250 MHz[8]	10GBASE-T Ethernet	Most commonly installed cable in Finland according to the 2002 standard. SFS-EN 50173-1
Cat.6a	U/FTP, F/UTP	500 MHz	10GBASE-T Ethernet	Adds outer shielding. ISO/IEC 11801:2002 Amendment 2.
Cat.7	F/FTP, S/FTP	600 MHz	Telephone, CCTV, 1000BASE-TX in the same cable. 10GBASE-T Ethernet.	Fully shielded cable. ISO/IEC 11801 2nd Ed.
Cat.7a	F/FTP, S/FTP	1000 MHz	Telephone, CATV, 1000BASE-TX in the same cable. 10GBASE-T Ethernet.	Uses all four pairs. ISO/IEC 11801 2nd Ed. Am. 2.
Cat.8.1	F/UTP	1600-2000 MHz	Telephone, CATV, 1000BASE-TX in the same cable. 40GBASE-T Ethernet.	In development.
Cat.8.2	F/FTP, S/FTP	1600-2000 MHz	Telephone, CATV, 1000BASE-TX in the same cable. 40GBASE-T Ethernet.	In development.

Fig. 14 – Cabluri torsadate utilizate in retele Ethernet.

Ca topologii se utilizeaza magistrala (bus – uzual cu cablu coaxial) si conexiunea de tip stea (star – uzual folosind cabluri bifilare torsadate).

Topologia de tip magistrala este un mediu de tip broadcast in care intarzaierile provin din propagare si repetoare (necesare pentru comunicatii la distanta pentru refacerea semnalului si mentinerea sincronizarii).

De ex., in fig. 15 se prezinta implementarea unei comunicatii la distanta necesara pentru accesul unei imprimante din departamentul B de catre un utilizator/statie de lucru din departamentul A.

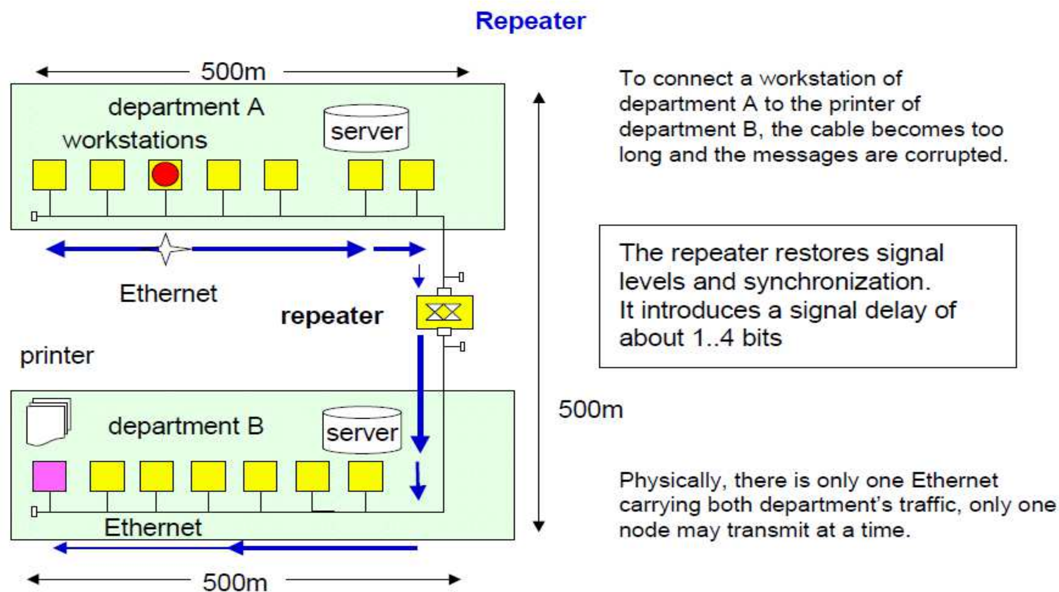


Fig. 15 – Utilizarea unui repetor penrtu conexiuni la distanta.

Repetorul introduce o intarzaiere de propagare de 1 pana la 4 biti.

Accesul la mediul de comunicatie se face utilizand mai multi algoritmi dintre care cel mai cunoscut este CSMA – CD (Carrier Sense Multiple Access with Collision Detection) – Ethernet half-duplex. Fiecare dintre algoritmi gestioneaza situatiile in care mai multi utilizatori (statii) acceseaza mediul de comunicatie (emit mesaje) in acelasi timp (aparitia coliziunilor).

Initial transmisia se facea dupa bunul plac al fiecarui emitator. Daca nu sosea o confirmare de primire, atunci mesajul este retransmis. Principiul a fost imbunatatit mai intai prin sesizarea de catre emitatori a starii canalului de comunicatie (Carrier Sense) evitand astfel sa transmita cat timp canalul este ocupat. O alta imbunatatire consta in detectia coliziunilor (CD). La detectia unei coliziuni dispozitivul poate transmite un semnal special in continuare pe durata cadrului de date pentru a se asigura ca toate dispozitivele au sesizat producerea coliziunii sau isi poate inceta transmisia imediat. Reluarea transmisiei se face dupa un interval aleator de timp, eventual din ce in ce mai mare (protocol nedeterminst).

Avantaj: Arbitrajul (pentru castigarea magistralei – canalului de comunicatie) nu depinde de numarul de competitori sau de adresa statiilor.

Dezavantaj: Nu exista un timp limita de asteptare. Acesta depinde de rata de sosire a cadrelor si de lungimea medie a acestora (protocol nedeterminst).

Obs:
– Desi protocolul este nedeterminist, in cazul unei incarcari usoare nu exista intarzaieri semnificative.
– Ethernet nu este considerata eficienta pentru cadre de dimensiuni mici si un numar mare de emitatori care emit simultan (fig. 16).
– Coliziunile nu pot fi detectate in integralitatea lor. In fig. 17 se prezinta probabilitatea pierderii unui pachet in functie de numarul de emitatori care emit simultan.

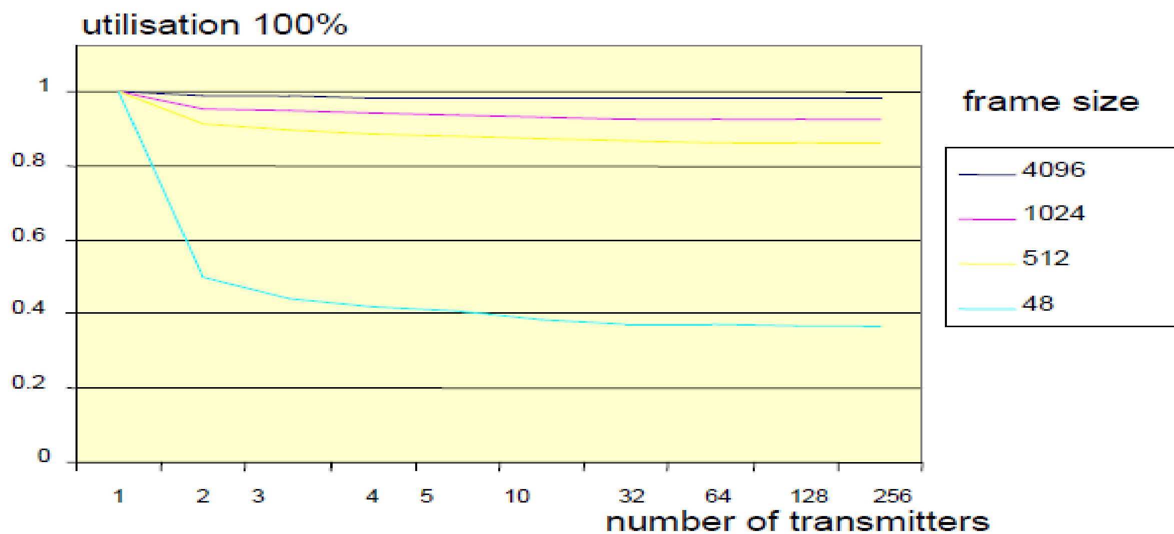


Fig. 16 – Eficienta retelei Ethernet.

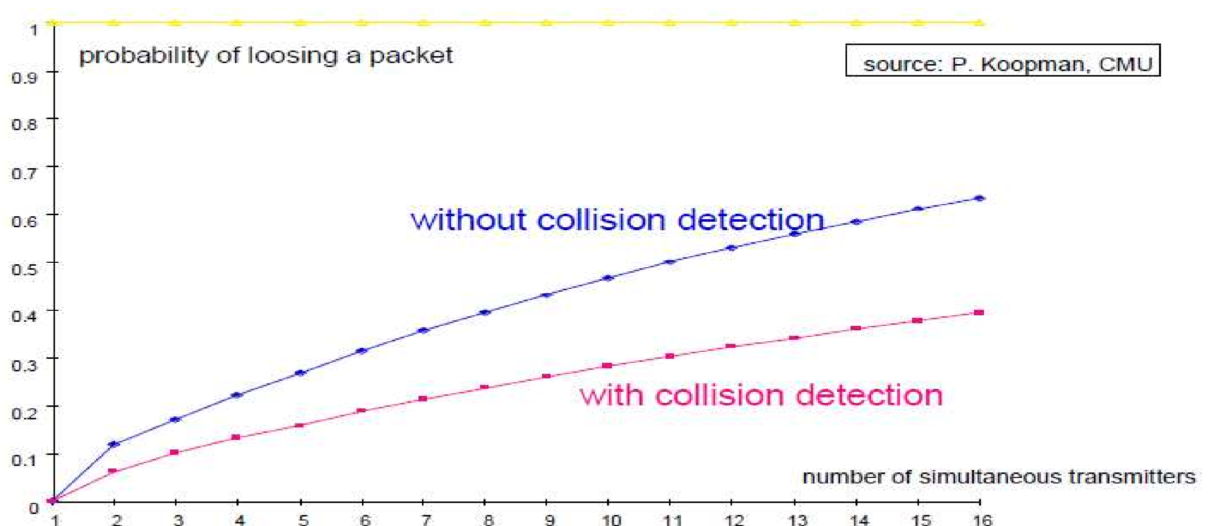
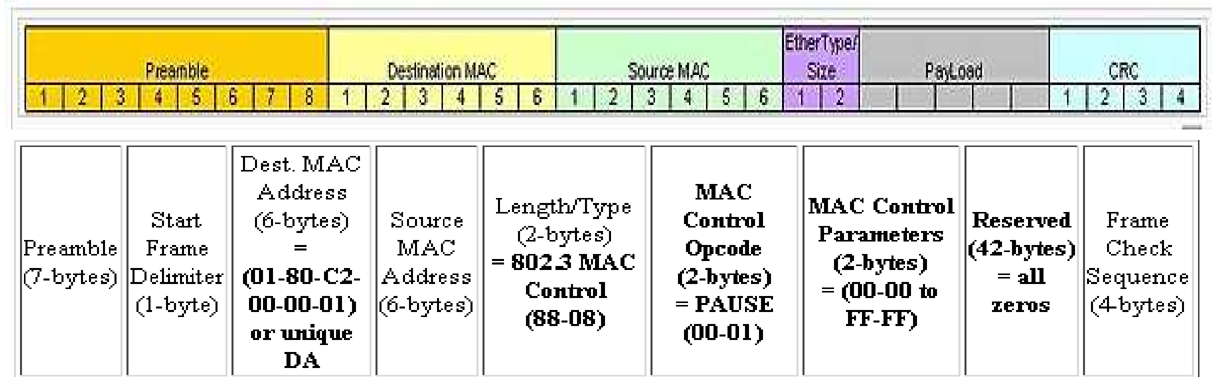


Fig. 17 – Probabilitatea de pierdere a unui pachet.

In fig 18 se reprezinta structura unor cadre Ethernet,



Preamble (7-bytes)	Start Frame Delimiter (1-byte)	Dest. MAC Address (6-bytes) = **(01-80-C2-00-00-01) or unique DA**	Source MAC Address (6-bytes)	Length/Type (2-bytes) **= 802.3 MAC Control (88-08)**	**MAC Control Opcode (2-bytes) = PAUSE (00-01)**	**MAC Control Parameters (2-bytes) = (00-00 to FF-FF)**	**Reserved (42-bytes) = all zeros**	Frame Check Sequence (4-bytes)

Fig. 18 - Structura unui cadru Ethernet. Preambulul foloseste pentru sincronizare la nivel de cadre.

iar in fig. 19 se prezinta cateva cateva interfete tipice (modalitati de conectare a statiilor la linia/canalul de comunicatie).

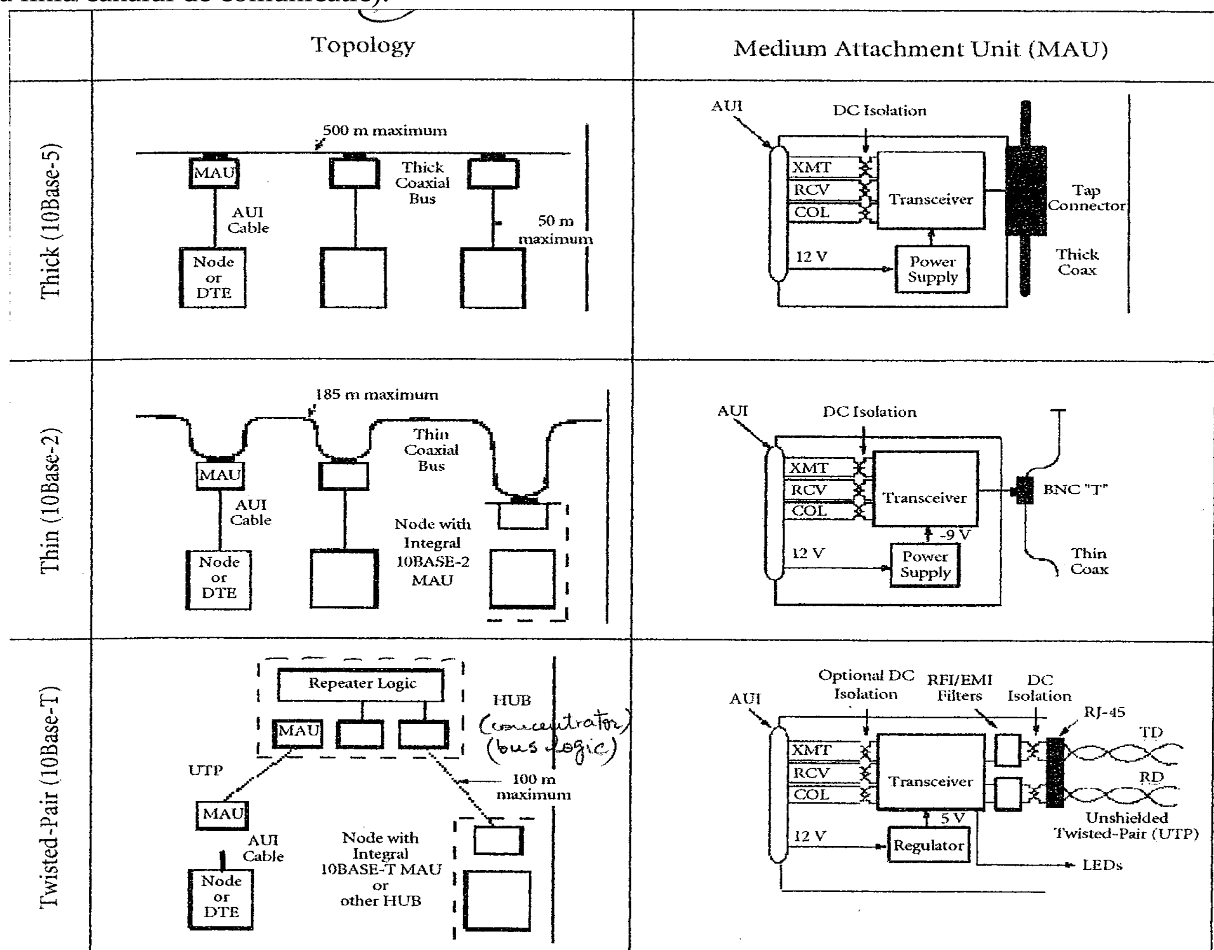


Fig 19 – Topologii si acces la mediul de comunicatie

<u>Obs</u>: In general, in cazul unei configuratii de tip stea, pentru conectarea unei statii (calculator sau alt dispozitiv similar – host N(etwork)I(nterface)C(ontroller)) la un hub (router, switch) este necesar un cablu direct (straight through cable). Pentu conectarea a doua dispozitive de acelasi tip/a doua porturi cu acceasi configuratie: MDI (Medium Dependent Interface) cu MDI (un calculator cu alt calculator) sau MDI-X cu MID-X (hub sau switch) este necesar un cablu de tip crossover. Pentru tehnologii mai noi (Auto MDI-X) cablul de tip crossover nu mai este necesar detectia si configurarea legaturii facandu-se automat. In fig. 20 se prezinta conexiunile in cazul unui cablu de tip crossover.

Pin	Connection 1: T568A			Connection 2: T568A Crossed			Pins on plug face
	signal	pair	color	signal	pair	color	
1	BI_DA+	3	white/green stripe	BI_DB+	2	white/orange stripe	
2	BI_DA-	3	green solid	BI_DB-	2	orange solid	
3	BI_DB+	2	white/orange stripe	BI_DA+	3	white/green stripe	
4	BI_DC+	1	blue solid	BI_DD+	4	white/brown stripe	
5	BI_DC-	1	white/blue stripe	BI_DD-	4	brown solid	
6	BI_DB-	2	orange solid	BI_DA-	3	green solid	
7	BI_DD+	4	white/brown stripe	BI_DC+	1	blue solid	
8	BI_DD-	4	brown solid	BI_DC-	1	white/blue stripe	

Fig. 20 –Conexiuni incrucisate.

<u>Tema</u>:

1. Repartizati semnalele la capatul nr. 2 al conexiunii pentru a realiza o legatura directa (dispozitiv la switch).
2. Determinati cu ajutorul unui tester tipul unui cablu ethernet (direct sau crossover).

3. Solutii de sisteme distribuite bazate pe aplicatii centralizate (Embedded Networking).

In acest tip de solutii distribuite controlul echipamentelor (de la distanta)se face de pe o statie de lucru prin intermediul unei aplicatii (interfete) complexe.

Caracteristici:

1.1. Procesoarele (sistemele incorporate) sunt apropiate de procesul controlat (comandat) si contin codul (programele) de control;

1.2. Monitorizarea se face de la distanta folosind PC-uri sau alte echipamente.
1.3. Intrefata grafica utilizator (GUI) este scrisa sa ruleze pe PC sub controlul unui anumit sistem de operare. Aceste interfete nu sunt in mod uzual standardizate. Trebuie de asemenea asigurata compatibilitatea driverelor de retea de la cele doua capete: supervizor si sistem incorporat. Din aceste cauze este dificila conectarea mai multor sisteme de acest tip intre ele.
1.4. Intretinerea si upgrade-ul se face in mai multe locuri: la fiecare statie de lucru si la fiecare sistem incorporat.
1.5. Interoperabilitatea GUI cu sistemul incorporat este foarte importanta. De multe ori o schimbare a codului aplicatiei (de pe sistemul incorporat – construit de ex. in jurul unui microcontroler) poate necesita de asemenea schimbari in GUI.

<u>Ca exemplu</u> de aplicatie pentru un astfel de sistem se prezinta controlul de la distanta a ecranului unui releu programabil Moeller/Eaton. Releul folosit este un Easy 822-DC-TC care nu dispune de port Ethernet. Ca urmare se foloseste un dispozitiv extern (Easy209-SE) care reprezinta o interfata intre portul RS-232 al relelului si reteaua Ethernet. Pe sistemul central (centrul de comanda) se afla in principal doua categorii de programe:

1. programe de depistare a dispozitivelor si initializare/configurare a interfetei de comunicare cu dispozitivele din retea (EASY 209-SE Configurator). In fig. 21 se prezinta un ecran de configurare pentru dispozitivul folosit.

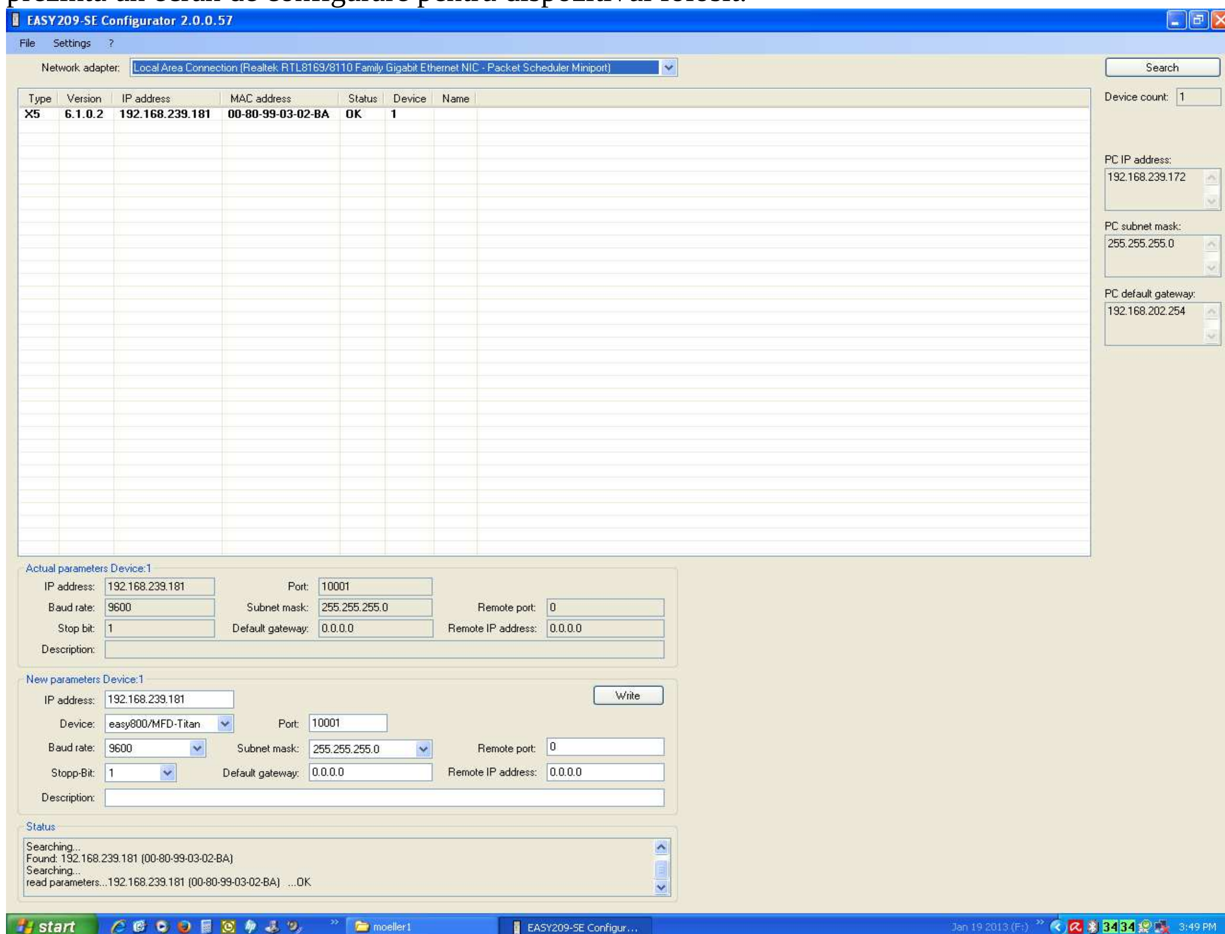


Fig. 20 – Detectie si configurare releu de la distanta.

Informatia este afisata in ecran astfel:
- In partea din stanga-sus sunt listate dispozitivelele descoperite in retea.
- In partea din dreapta-sus sunt listati parametrii de retea ai PC-ului din centrul de comanda.
- In partea de jos sunt listati parametrii actuali ai dispozitivului descoperit si selectat si respectiv parametrii noi care trebuie programati in dispozitiv.

2. programe de control/comanda (EASY SOFT).

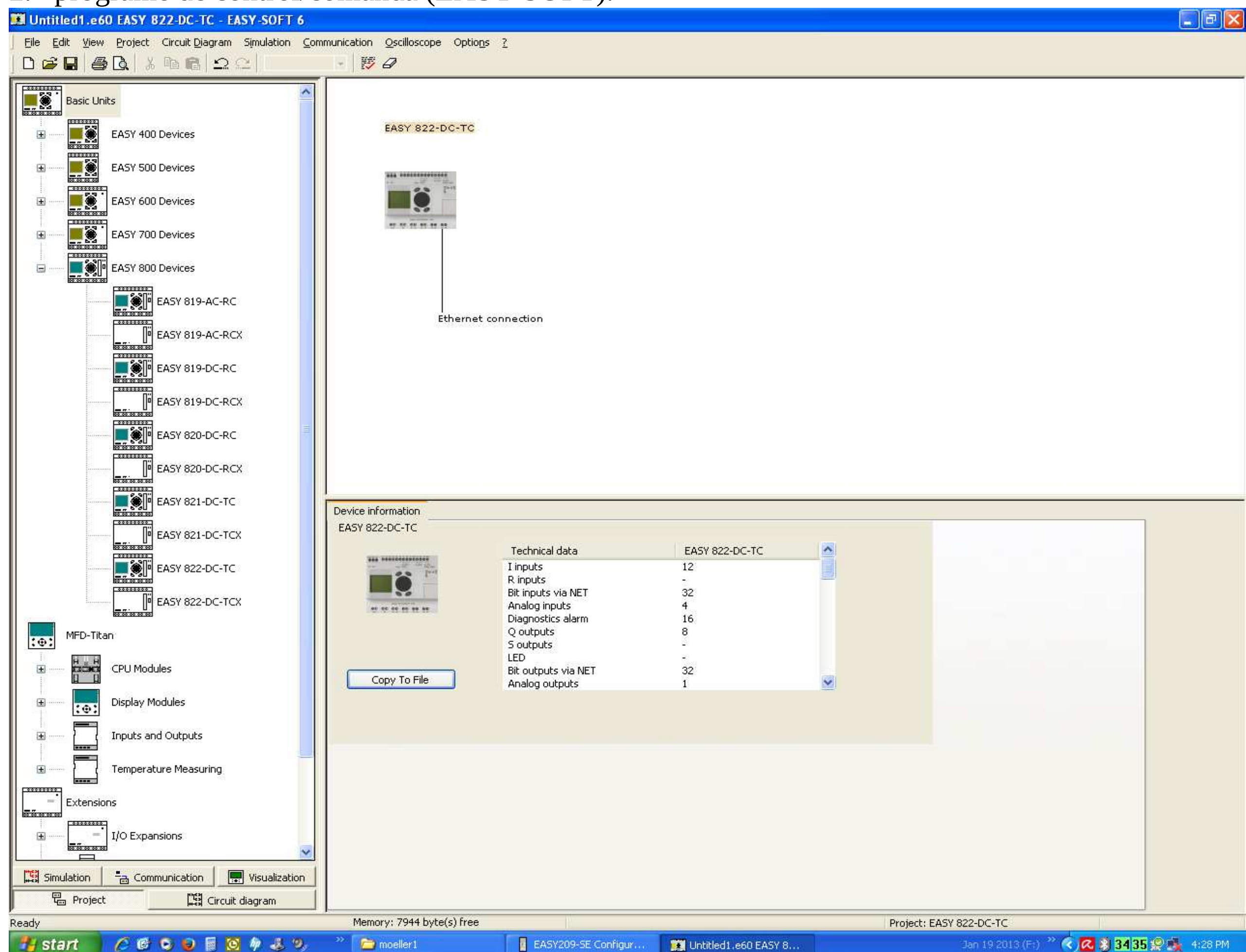


Fig. 22 – Controlul de la distanta al unui releu Moeller folosind aplicatia EASY SOFT. Alegerea dispozitivului si a interfetei de conectare la dispozitiv.

2.1. Dispozitivul se face cunoscut intefetei prin mutarea/glisarea acestuia in fereastra din dreapta sus dupa ce este selectat din lista de dispozitive din fereastra din stanga-sus. Selectia interfetei de conectare cu dispozitivul se face prin bifarea casutei ‘Ethernet Connection’ din fereastra de jos.

2.2. Stabilirea unei legaturi cu dispozitivul (aflat la distanta) se face in urmatorii pasi:

2.2.1. Se apasa butonul ‘Communication’ din stanga jos.

2.2.2. Se apasa butonul ‘Connection’ din stanga sus a ecranului nou deschis (Fig. 23). Obs: Interfata de comunicatie si parametrii acesteia (IP, Mask, etc.) pot fi modificati editand profilele Ethernet din partea din stanga sus a ecranului.

2.2.3. Se apasa butonul ‘On-line’ si se asteapta conectarea (Butonul Off-line devine activ).

2.2.4. Se apasa butonul ‘Display’ si se selecteaza optiunea ‘Display & buttons’.

2.2.5. In partea de jos a ecranului apare exact ecranul relelului Moeller de la distanta.

2.3. Testarea conexiunii de poate face fie adresand comenzi (apasand butoanele) ecranului tocmai afisat pe ecranul PC-ului prin intermediul mouse-ului (transfer informatie de la PC la releu) si observand modificarea ecranului releului, fie adresand comenzi (apsand butoanele) releului si observand modificarea ecranului releului afisat pe ecranul PC-ului.

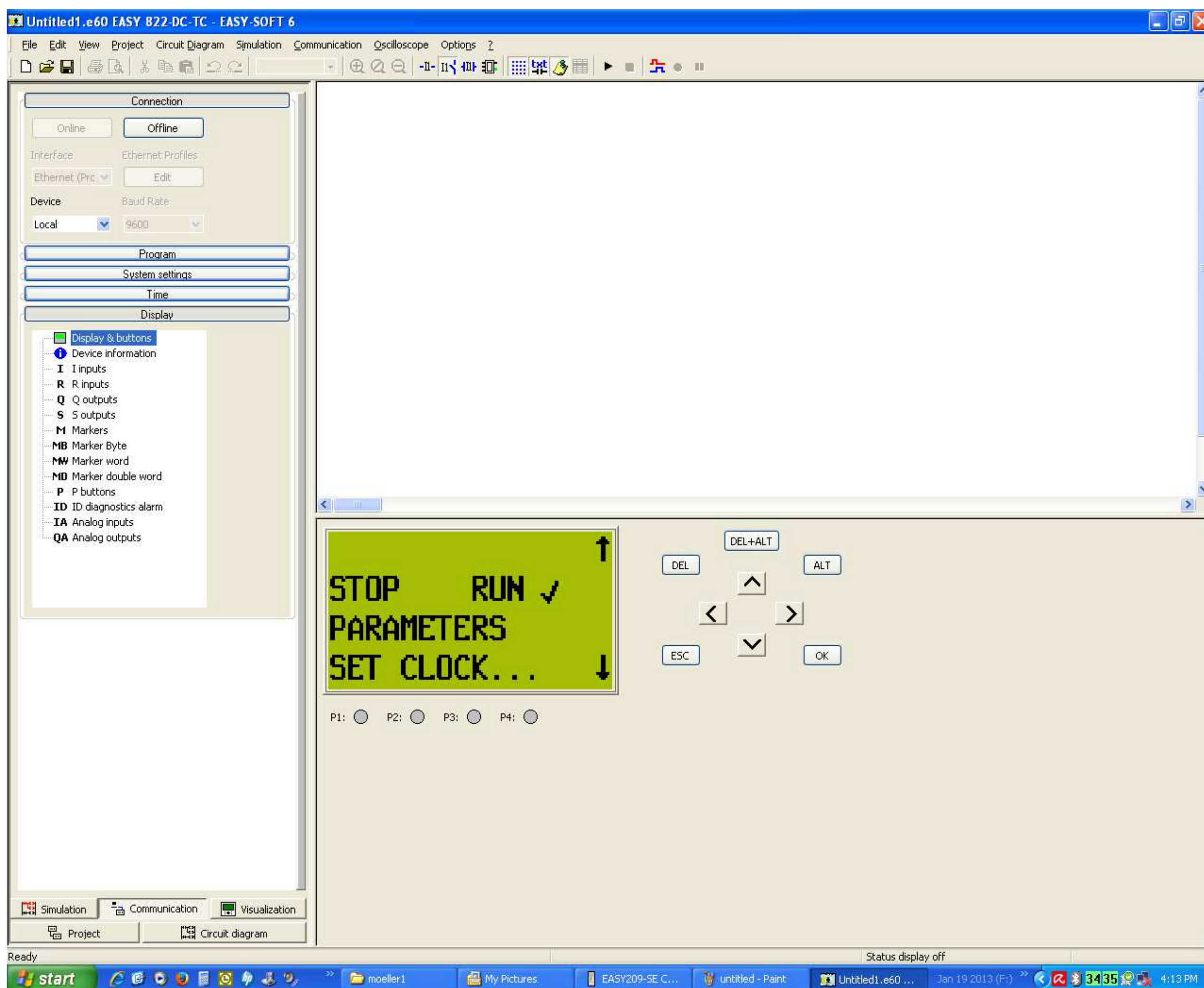


Fig. 23 – Controlul de la distanta al unui releu Moeller folosind aplicatia EASY SOFT. Transmisie de date intre PC si releu.

Obs: Se pot de asemenea afisa intrari/iesiri digitale, intrari/iesiri analogiceprecum si alti parametri ai releului si se poate programa de la distanta releul.

4. Solutii descentralizate controlate de oriunde (Embedded Internet).

Multe dintre sistemele incorporate (embedded systems) sunt folosite in aplicatii de tip control-comanda in sensul sesizarii unor evenimente externe si comenzii asupra unor procese locale sau la distanta. Multe dintre aceste sisteme , bazate pe microcontrolere, sunt administrate de operatori umani prin intermediul unor interfete om-masina. Exemple: casierie, telefon mobil, ecranul unui televizor, o interfata PC.

In cele mai multe cazuri aceasta interfata om-masina este cea care este cea mai costisitoare din punctul de vedere al timpului si banilor necesari pentru dezvoltarea acesteia.

Totodata din ce in ce mai multe sisteme incorporate sunt interconectate in general prin intermediul unor interfete seriale (UART, CAN, SPI, Ethernet). Aceste standarde hardware trebuie sustinute de standarde software aflate de multe ori la 'mana' inginerului de sistem. De multe ori este posibil, din aceasta cauza, ca doua sisteme bazate pe o interfata (hardware) de tip UART sa nu se poata interconecta desi au la baza acelasi standard hardware.

Intreconectarea sistemelor incorporate prin intermediul retelei Internet incearca sa rezolve aceste doua probleme: cea a conectivitatii universale (eliminand astfel discordanta intre diferitele protocoale de interconectare (de retea)) si cea a interfetei om-masina (eliminand astfel inconsistenta interfetelor de tip utilizator), oferind ca standard universal de interconectare protocolul TCP/IP, iar ca administrator la interfetei om-masina popularul browser de internet.

Caracteristici:

1. Prin intermediul protocolului TCP/IP datele sunt transmise prin reteaua internet de la o adresa (IP) la alta. Aceste protocoale stau la baza altor protocoale la nivele superioare ale arhitecturii OSI cum ar fi:
   1.1. FTP: pentru transferul fisierelor de la un calculator la altul.
   1.2. HTTP: pentru trimiterea unei pagini web de la un server web la un browser.
   1.3. HTTPS: la fel ca mai inainte, dar intr-un mod securizat (criptat).
   1.4. POP3: receptionare de mesaje e-mail.
   1.5. SMTP: emitere de mesaje e-mail.
2. In unele cazuri se foloseste protocolul UDP in locul protocolului TCP.
3. Protocolul hardware este in general Ethernet, dar poate fi si altul (Bluetooth, DSL, Cable, etc.) compatibil cu protocolul TCP/IP. Ca viteze de transmisie se utilizeaza in special cea de 10Mbps (standard 10Base-T) – suficienta de obicei in lumea sistemelor incorporate uzuale plasate in spatele unui switch sau a unui ruter care sunt repsonsabile cu trimiterea ocazionala (cu frecventa relativ mica) a unor rafale de date – dar si cea de 100Mbps (standard 100Base-T) pentru aplicatiile de mare performanta (legaturi voiceIP, sisteme de navigare in industria auto, etc.).

4. Ethernet-ul devine din ce in ce mai popular si in lumea sistemelor incorporate (embedded systems). De la procentul de 8% detinut in 1998 acesta a ajuns la 24% in 2004 in timp ce alte magistrale au crescut mult mai putin (Profibus, Device Net, Fieldbus).
5. Problema trimiterii datelor in timp real in mod determinist si cu intarzaiere minima a datelor ramane (a se vedea alte tipuri de retele Ethernet de ex. industrial Ethernet) desi unele solutii bazate pe switch-uri si rutere au fost deja propuse si utilizate.
6. Incorporarea de facilitati de comunicare Ethernet si Internet se loveste de cateva dificultati in lumea sistemelor incorporate bazate pe microcontrolere: pret, complexitate si dimensiune, memoria interna (de program) a microcontrolerelor. Aceste bariere pot fi insa depasite in mai multe faze. Astfel **solutia Microchip** a presupus urmatoarele etape:
   6.1. Prima faza a fost cea a adaugarii unui controler Ethernet (100 pini) ‘imprumutat’ din lumea PC-urilor bazat pe o interfata ISA (din aceasta fiind folositi 16 pini de date, 5 de adresa si 2 semnale de comanda pentru citire-scriere), ceea ce a dus la cresterea complexitatii sistemului, a spatiului ocupat pe placa si a necesitatii folosirii unui microcontroler cu un numar mare de pini (80) – Fig. 24.

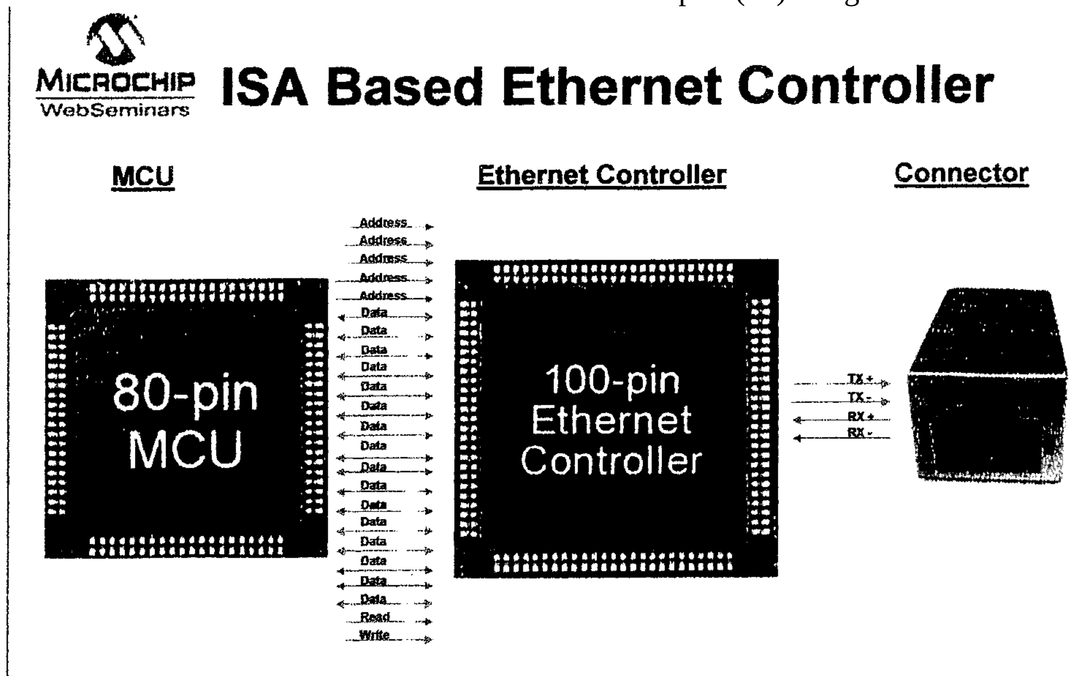


Fig. 24 – Solutia de tip PC.

   6.2. A doua faza a constat in proiectarea unui controler de retea care sa comunice cu microcontrolerul pe un numar mult mai redus de pini decat magistrala ISA. S-a ales solutia magistralei seriale SPI (numai 4 fire). Aceasta solutie a dus la posibilitatea folosirii unui microcontroler cu un numar mai redus de pini (64), la reducerea costului cu 34% si a spatiului folosit cu 52% - Fig. 25.
   6.3. A treia faza este ce a integrarii controlerului de retea in pastila microcontrolerului (de ex. PIC18F67J60 cu 64 de pini) ceea ce a dus la o reducere a costului cu 51% si a spatiului ocupat cu 68% fata de solutia din prima etapa – Fig. 26.

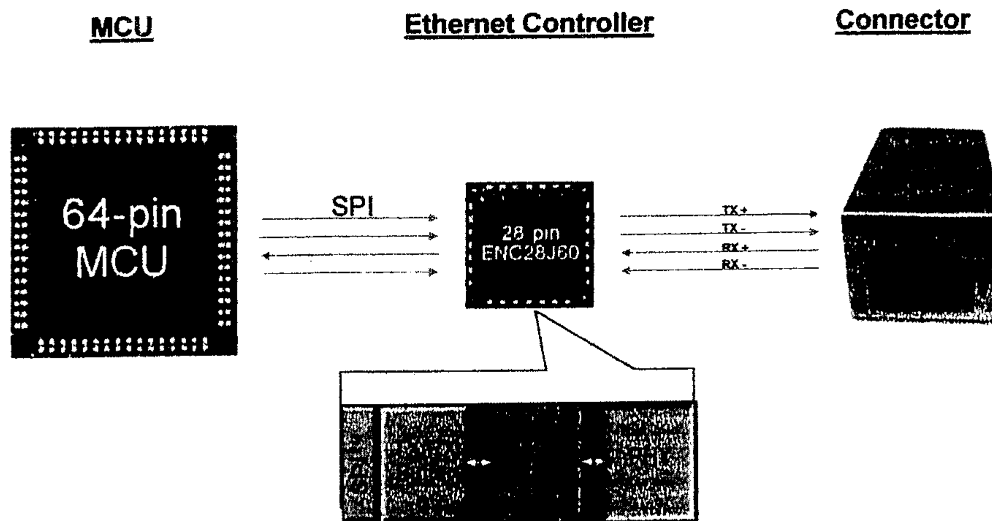


Fig. 25 – Controler Ethernet separat.

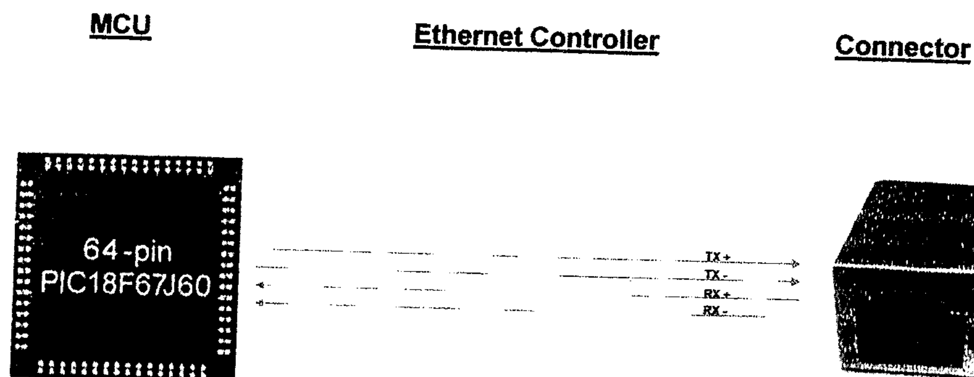


Fig. 26 – Controler Ethernet integrat (port Ethernet la nivel de microcontroler).

Ca exemple de aplicatii se prezinta in Fig. 27 schema bloc principiala a controlului camerei unui hotel,

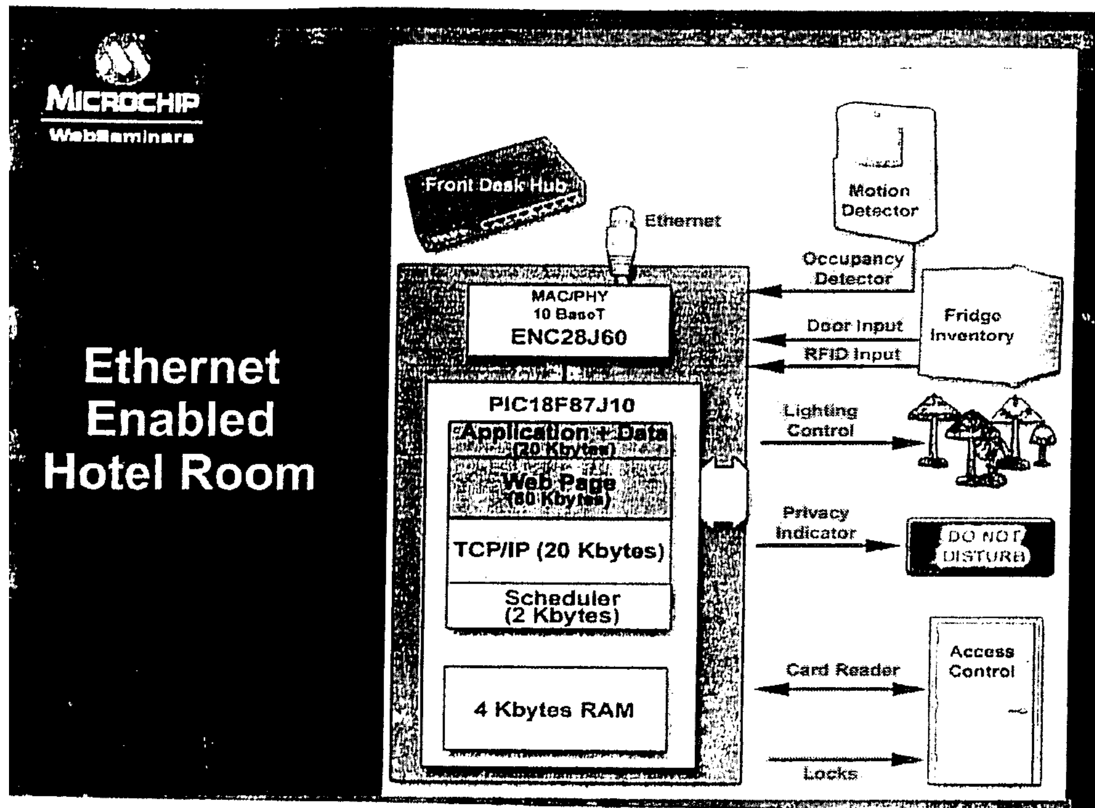


Fig. 27. Camera de hotel controlata prin Internet.

iar in Fig. 28 solutia Microchip pentru controlul de la distanta a unui automat de produse.

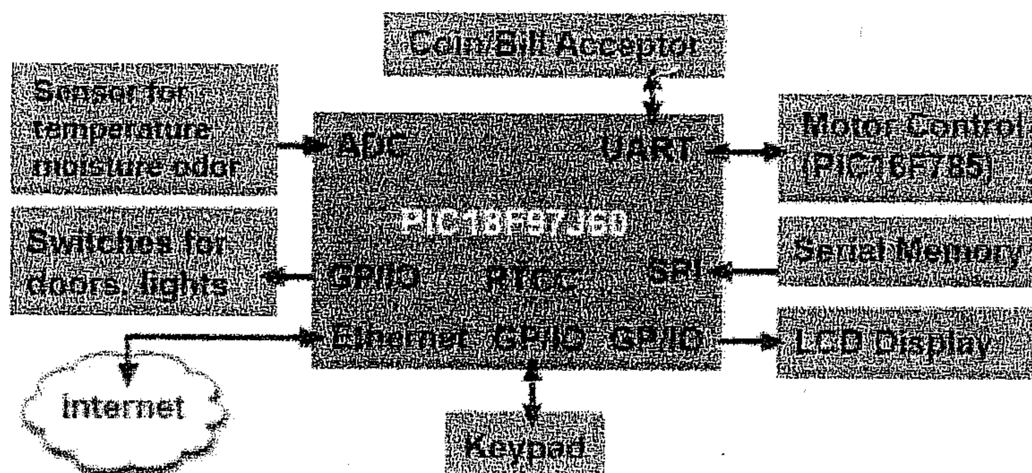


Fig. 28 – Automat de produse contrlat prin Internet.

7. Ca solutii de integrare software a sistemelor incorporate conectate in retea amintim 3 solutii:
   7.1. Solutia Pseudo TCP/IP stack - utila pentru retele mici de cost redus cu facilitati de control erori reduse. Solutia necesita software si hardware scump la nivel de gateway pentru interconectarea cu alte sisteme si necesita cel putin doua puncte de intretinere software: microcontroler si gateway.
   7.2. Solutia separata – in care stiva TCP/IP este achizitionata ca software separat. Se asigura un grad mare de integrare, un singur punct de mentenanta software (la nivelul microcontrolerului), dar necesita asigurarea interoperabilitatii intre toate componentele (API, cod aplicatie, drivere, eventual RTOS **si** stiva TCP/IP) in conditiile unei curbe de invatare destul de lenta (in ceea ce priveste produsul achizitionat) si a cerintelor si asistentei acordate de cel care a livrat stiva TCP/IP.
   7.3. Solutia integrata cu hardware si software de retea intr-un singur produs (in care si partea de retea este proiectata si realizata de cel care proiecteaza si (unele) dintre celelalte module – vezi solutia Microchip, Texas Instruments, etc.).
8. Un (micro)controler care suporta (contine cod si pentru) protocoalele TCP/IP (TCP/IP stack) si un protocol de tip HTTP se transforma intr-un Server Web. Paginile web inmagazinate de obicei intr-o memorie EEPROM, pot fi transmise, prin intermediul pachetelor TCP/IP, pe o conexiune Ethernet catre un computer care formuleaza cereri in acest sens prin intermediul unui browser.
9. Daca pe stiva TCP/IP este construit si protocolul SMTP si daca reteaua are acces la un server de e-mail, atunci tehnicianul (sau oricare alta persoana interesata) poate fi atentionat de sistemul incorporat (microcontroler) in legatura cu anumite evenimente prin intermediul mesajelor de e-mail.
10. Browserul – reprezinta un standard in prezentarea documentelor. Sarcina sa principala este preluarea unui document (de ex. preluat de pe un server web) format dintr-un text formatat cu ajutorul unui limbaj special (HTML) si prezentarea acestuia pe un dispozitiv de afisare, uzual un display. Exista astfel un GUI independent de platforma de calcul (si cu atat mai mult fata de sistemul de calcul), usor de inteles si de utilizat.
11. De asemenea utilizatorul, sub controlul browserului, poate trimite date spre serverul web prin intermediul unor ‘formulare’ sau butoane (vezi de ex. protocoalele GET si POST), aceste date fiind receptionate de ‘stiva’ TCP/IP (TCP/IP stack) de pe serverul de web si trimise la programul din memoria interna a microcontrolerului care controleaza functionarea sistemului de la distanta in acelasi mod cum sunt transmisi parametrii de catre programul principal unei subrutine (functii). Astfel poate fi, de ex., activat un motor sau o cerere de date de la o retea de senzori.

Avantaje:

1. Fiabilitate sporita a sistemului distribuit prin monitorizarea continua si achizitia de date in vederea prezicerii defectiunilor care pot aparea.
2. Diagnoza si mentenanta imbunatatita prin monitorizarea, service-ul si mentenanta de la distanta prin intermediul legaturii TCP/IP. Singurele puncte de mentenanta software sunt cele de la nivelul sistemului incorporat (microcontrolerului). Codul poate fi schimbat de la distanta folosind orice browser cu drepturi de securitate

corespunzatoare. Costul administrarii retelei este mult diminuat, iar mentenanta este mult simplificata.
3. Optimizarea productiei prin monitorizarea tuturor aspectelor importante din procesul de productie: incarcarea masinilor, produsele livrate la iesire, ‘gatuiri’ in procesul de productie, etc.

Dezavantaje:
1. Securitate.
2. Livrarea datelor in timp real.

Ca exemplu se considera controlul de la distanta al unui microsistem bazat pe microcontroler Microchip – PICDEMNET (Fig. 29).

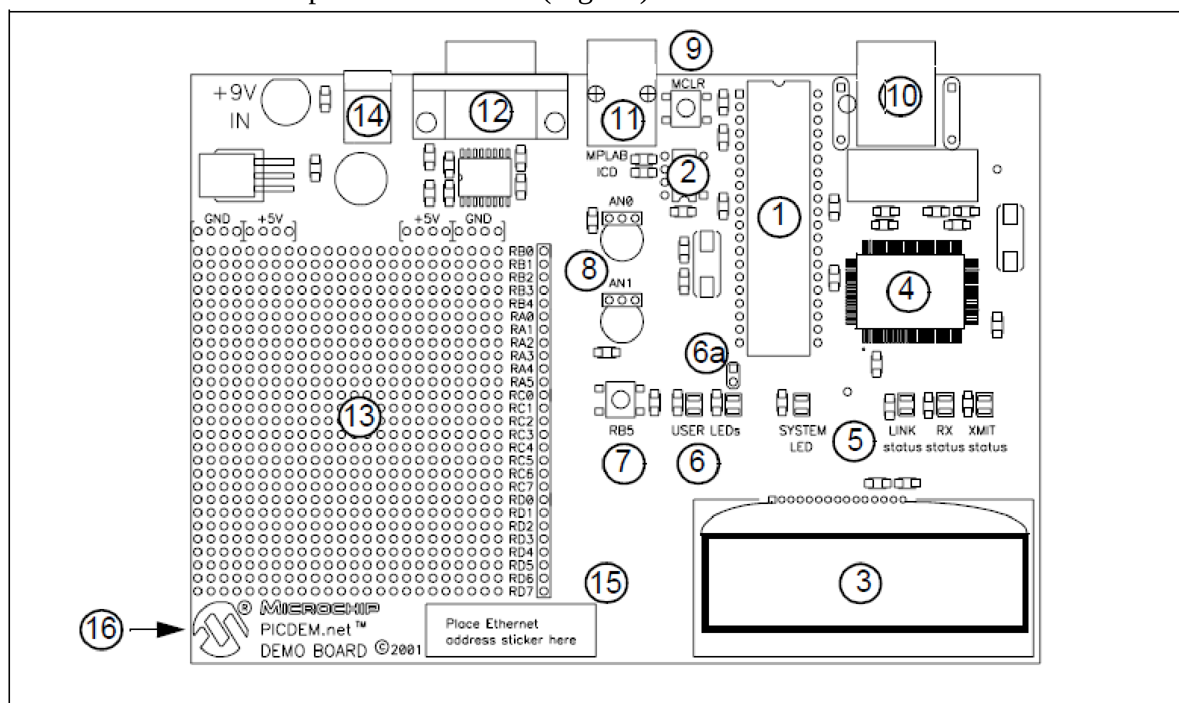


Fig. 29 – Placa PICDEMNET.

1. Soclul microcontrolerului – Din fabrica soclul este populat cu un microcontroler PIC18F452 in care este programata aplicatia demonstrativa. Alte microcontrolere compatibile: PIC16F877, PIC18C452 si PIC18F458.
2. Memoria EEPROM – de 256Kbiti (32KB) pentru memorarea paginilor Web. Memoria este programabila prin intermediul unei legaturi seriale I^2C.
3. Display LCD.
4. Controler Ethernet (RealTek RTL8019AS)
5. Leduri de stare – in numar de 4.
   5.1. SYSTEM – clipeste pentru a arata ca placa este alimentata si ca este conectata corect.
   5.2. LINK STATUS –indica daca conexiunea Ethernet este activa (cand este aprins).
   5.3. XMIT si RX – indica transmiterea sau respectiv receptia unui pachet de date.

6. Leduri utilizator – doua leduri controlate prin intermediul unor semnale (pini) I/O digitale. Simuleaza o iesire digitala pentru controlul unui proces.
7. Buton utilizator – conectat la un pin I/O numeric al microcontrolerului. Simuleaza o intrare digitala de la un proces.
8. Potentiometre utilizator – simuleaza doua intrari analogice generate de un proces.
9. Butonul de RESET – pentru resetarea (reinitializarea) placii (sistemului).
10. Conector RJ-45 – pentru conectarea la retea (10-Base T).
11. Conector RJ-11 – pentru conectarea la un depanator.
12. Conector RS-232 de tip DB9M – permite configurarea sistemului folosind un port serial standard. De asemenea portul permite utilizatorului sa descarce noi pagini Web in memoria EEPROM a microsistemului.
13. Zona de proiectare (Prototype area) – pentru adaugarea de circuite noi la microsistem.
14. Sursa de alimentare – permite alimentarea de la surse de energie continuie sau alternative.
15. Identificator Ethernet – cei mai putin semnificativi 2 octeti ai MAC-ului placii (MAC = Media Access Control).

Monitorizarea si controlul sistemului se realizeaza folosind un browser standard (Internet Explorer, Mozilla Firefox), lansat in executie de pe un PC din reteaua la care este conectat microsistemul.

Obs:

1. Paginile Web pot fi (si sunt in general) create si optimizate pentru un anumit browser.
2. Este posibil ca pentru controlul microsistemului sa poata fi folosit orice dispozitiv care suporta browserul respectiv situat oriunde in lume, cu conditia asigurarii accesului la microsistemul respectiv (de ex. reteaua universitatii este protejata printr-un firewall/proxy care face dificil accesul din exteriorul acestei retele).

La introducerea adresei sistemului (IP) browserul cere de la sistem (prin intermediul serverului Web programat in microcontroler) o pagina Web (cea de start) - Fig. 30.

Fig. 30 – Pagina Web pentru monitorizarea si controlul microsistemului

Verficarea monitorizarii/controlului se face prin intermediul butoanelor (control – dreapta jos) si respectiv a zonei de afisare (stanga jos).

Astfel:

1. La apsarea butonului Toggle LED1 ledul 1 de pe placa isi schimba starea (aprins/stins).
2. La apsarea butonului Toggle LED2 ledul 2 de pe placa isi schimba starea (aprins/stins).
3. Indicatorul ‘Switch’ arata starea intrerupatorului utilizator de pe placa (1 – liber, 0 - apasat).
4. Indicatorul ‘LEDs’ arata starea ledurilor de pe placa (poate fi vazut ca un control al realizarii cu succes a sarcinii de comutare a starii celor doua leduri de pe placa/microsistem).
5. Indicatorul ‘Pot1’ indica valoarea tensunii generate prin intermediul potentiomentrului 1 de pe placa (poate lua valori in intervalul 0-1023).
6. Indicatorul ‘Pot2’ indica valoarea tensunii generate prin intermediul potentiomentrului 2 de pe placa (poate lua valori in intervalul 0-1023).